NETWORK-BASED SOFTWARE RECOVERY FOR COMPUTING DEVICES

BACKGROUND OF THE INVENTION

The invention relates generally to the field of computing devices and, more particularly, to methods and systems for providing recovery software to computing devices.

In a build to customer order (BTCO) computing device manufacturing environment, a custom software package is typically assembled for each individual computing device. In addition to loading the customized software package on an individual computing device, a recovery mechanism must also be developed in order to provide an image of the customized software package to the individual user. By providing this image of the installed software package, the individual user can reload the software package previously installed in the manufacturing environment when the need arises. Typical events that require users to reload the factory installed software package include rebuilding the package after a virus has corrupted an internal hard drive, after a head crash, or when the user purchases a new hard drive.

Computing device manufacturers expend significant effort in producing readable media for each individual, customized software package. These efforts include bearing the cost of the individual media, requiring substantial "touch" labor in order to load the software package onto the readable media, as well as managing the coordination activities required to ensure that a particular readable media is shipped along with the particular customized computing device. These costs are generally passed on to consumers.

Further, in non-BTCO manufacturing environments, where standardized products are produced with little variation in the configuration of the installed software package, an update to a certain portion of the software can require the manufacturer to discard a large number of units of the readable media that are in stock
5   when the updated software package becomes available. The costs of these discarded units, as well as the need to keep these media in stock, places an additional burden which the manufacturer must pass on to consumers.

Therefore, it is highly desirable to employ a method and system for performing software recovery which does not require the production of a customized
10   media for delivery along with the computing device. Such a method would lower the cost of BTCO computing devices as well as increasing manufacturing efficiency. The method and system would also reduce the need to discard readable media when a software update becomes available in non-BTCO manufacturing environments. These benefits can then be passed on to consumers in the form of reduced equipment
15   costs and increased availability of both BTCO and non-BTCO computing devices.

SUMMARY OF THE INVENTION

According to one aspect of the invention, a computing device performs a
20   method for receiving recovery software from a network. In the method, the computing device transmits an identifier to a network location. The transmitted identifier, which can be a serial number or other unique identifier that is indicative of a particular software package installed on the computing device, is received by a network server which determines the software package previously installed on the
25   computing device during the manufacturing of the device. The server can then assemble the particular software package and transmit the package to the individual computing device. The receiving computer device can then reload the factory-installed recovery software. Other software can also be transmitted to the user in a similar fashion.
30   The invention is pointed out with particularity in the appended claims. However, a more complete understanding of the various aspects of the invention may

be acquired by reading the description herein, in conjunction with the figures, wherein like reference numerals refer to like elements.

BRIEF DESCRIPTION OF THE DRAWINGS

5

FIG. 1 is a block diagram of a computing device and network server used in network-based software recovery for computing devices in accordance with a preferred embodiment of the invention;

FIG. 2 is a block diagram of a method used in network-based software

10   recovery for computing devices in accordance with a preferred embodiment of the invention.

FIG. 3 is a block diagram of a method used in a computing device which performs network based software recovery in accordance with a preferred embodiment of the invention; and

15   FIG. 4 is a block diagram of a method used in a network server which performs network based software recovery in accordance with a preferred embodiment of the invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT'S

20

Network-based software recovery for computing devices provides a low-cost and efficient means of conveying software to the computing device that was previously installed during the manufacturing of the computing device. The method is especially useful for BTCO computing devices where a customized software

25   package must be provided to an individual computing device and an image of the software package is typically loaded on an accompanying readable media, such as a compact disc, in order to enable the user to rebuild the custom software package after the integrity of the data on an internal disk drive has been compromised. Such compromising events include damage to data files by way of a computer virus, or

30   after the user has replaced the internal disk drive. The method is also useful in BTCO and other environments to enable the timely and efficient delivery of upgraded software when that software becomes available. Through the use of network-based

software recovery, the software package is transmitted to the computing device from a network location, such as a site on the World Wide Web, thus obviating the need to load the custom software package on a readable media for delivery to the user along with the computing device.

5   FIG. 1 is a block diagram of a computing device used in network-based software recovery for computing devices in accordance with a preferred embodiment of the invention. In FIG. 1, computing device 10, which can be any multi-purpose computing resource such as a laptop or desktop computer, multi-function pager, or personal digital assistant, is interfaced to network 20. Computing device 10 includes

10   network transceiver 30, which enables the device to transmit and receive messages with other computing devices which may be interfaced with network 20. Additionally, computing device 10 includes internal disk drive 60, which provides memory storage capability for the computing device, peripheral memory device 80, as well as removable memory media 85. Processor 50 provides coordination as well as

15   management of internal disk drive 60 and peripheral memory device 80, and network transceiver 30. Although internal disk drive 60 is most useful when computing device 10 is a laptop or desktop computer, nothing prevents the use of such a disk drive when computing device 10 is a hand-held device such as a multi-function pager, or personal digital assistant.

20   In a preferred embodiment, internal disk drive 60 represents an internal resource which comprises memory for a variety of application programs which can be executed by computing device 10. These programs include an operating system, internal and peripheral device drivers, and other programs, routines, and symbolic languages that control the functioning of the computing device 10 and direct its

25   operation. It is anticipated that internal disk drive 60 embodies a media which can be written to and read from while processor 50 is executing any one of several programs. Further, the media of internal disk drive 60 typically rotates about a central axis during most operations of processor 50. Under these conditions, internal disk drive 60 represents a memory media which is vulnerable to viruses or other destructive

30   software encoded within data files accessed by any one the several programs. Additionally, internal disk drive 60 is also vulnerable to mechanical failures due to the

constant movement of the media and the need to maintain stationary read and write heads in close proximity with the media of internal disk drive 60.

In the event that internal disk drive 60 becomes corrupted, either through the action of a computer virus or errant software, or if internal disk drive 60 is replaced

5   either due to a hardware failure or merely the user's desire to upgrade to higher capacity disk, the user can load removable memory media 85 into peripheral memory device 80 for access by processor 50. Desirably, removable memory media 85 includes a basic bootstrap and browser capability that allows the computing device 10 to access network 20 and communicate with network server 100 without requiring

10   substantial user interaction. In a preferred embodiment, this user interaction is expected to be limited to a user inputting the serial number or other identifier which is an attribute of the particular computing device. Alternatively, computing device 10 may require no user input in order to transmit an identifier across network 20 to network server 100. In this embodiment, read-only memory 90 (which can be an

15   EEPROM other non-volatile storage media) can store the identifier of the particular computing device, which also eliminates the need for the user to enter the identifier by way of keyboard input device 40.

In a preferred embodiment, the software loaded onto removable memory media 85 does not include an image of the custom software package loaded on

20   internal disk drive 60 during the manufacturing process. Rather, removable memory media 85 includes standardized software which serves to establish a link between computing device 10 and network server 100 by way of network 20. Additionally, the removable memory media 85 include the software to communicate an identifier from keyboard input device 40, removable memory media 85, or read-only memory 90 to

25   network server 100.

Alternatively, removable memory media 85 can be loaded with a standard portion of the software package installed during manufacturing, thus requiring that only the customized portion of the software package be conveyed to the particular computing device during the software recovery process. For example, in the event

30   that all models of computing devices, which are similar to computing device 10, are loaded with a single version of graphical user interface software, this standard software can be loaded onto removable memory media 85, thus requiring the

6

download of only a substantial portion of the software package installed during manufacturing of computing device 10.

When the identifier from computing device 10 is received a at network server 100, which resides at a particular network location, network transceiver 110 receives

5   this identifier and conveys the identifier to processor 150. Processor 150 then accesses database 130 in order to determine the particular software package loaded on computing device 10 during manufacturing. When processor 150 ascertains the particular software package to which computing device 10 is entitled, repository 140 is accessed in order to begin assembling and transmitting the recovery software

10   package to computing device 10. These files are then conveyed through network server 100 and on to computing device 10 by way of network 20.

When database 130 is accessed in order to determine the particular software package loaded on computing device 10, database 130 can include an indicator which denotes that an upgrade to a certain portion of the previously installed software

15   package is available. Thus, in the event that defects in the particular software entity have been corrected and that a version of the software entity is now available, this updated version of the software can be conveyed to the user, or perhaps even offered for sale to the user. In the event that enhancements have been made to increase a capability of the software entity, these enhancements can be offered to the user as

20   well. This allows network server 100 to offer the most advanced and up-to-date versions of the software package to the user of computing device 10.

Alternatively, database 130 can be used to offer a stable version of the previously installed software package. For example, an enterprise customer may have a need to ensure that all computing devices of a certain type maintain a specific

25   software package configuration in order to reduce information technology support costs. Thus, database 130 can be queried in order to ascertain the precise software version installed on the particular computing device during manufacturing. Repository 140 can then be accessed in order to begin assembling and transmitting the recovery software package to computing device 10. These files are then conveyed

30   through network server 100 and on to computing device 10 by way of network 20.

In FIG. 1, network 20 is anticipated to be any type of network which is capable of transferring digital information between a number of computer resources,

such as computing device 10 and network server 100. Thus, network 20 may be a portion of the Internet which allows connectivity among a large number of computing resources distributed throughout the World Wide Web. In a similar manner, network 20 can also be representative of a smaller, more private network such as those which

5   exist within many enterprise customers' information distribution networks. Further, access to network 20 by computing device 10 and network server 100 may be managed by a service provider which facilitates information transfer between the computing device and the network server by way of intervening computer resources not shown in FIG. 1.

10   FIG. 2 is a block diagram of a method used in network-based software recovery for computing devices in accordance with a preferred embodiment of the invention. The apparatus of FIG. 1 is suitable for performing the method of FIG. 2. In block 200, a computing device transmits an identifier to a network location. This transmission can be facilitated by way of a local area network or a global

15   communications network, such as the Internet. In block 210 a network server at a network location receives the identifier. In block 220, the network server determines the software package previously installed during the manufacturing of the computing device. Block 220 additionally includes the transmission of the software package to the computing device. Block 220 can also include determining if an updated or

20   upgraded version of a certain portion of the software package is available. At block 230, at least a portion the software package is received by the computing device.

         FIG. 3 is a block diagram of a method used in a computing device which performs network based software recovery in accordance with a preferred embodiment of the invention. Computing device 10, operating in conjunction with

25   keyboard input device 40, peripheral memory device 80 and removable memory media 85, are suitable for performing the method of FIG 3. Additionally, the method steps of FIG. 3 can be tangibly embodied using a removable memory media (such as removable memory media 85) that includes a program of instructions executable by computing device 10 in order to perform the method steps for directing the computing

30   device to access a network location for receiving a software package previously installed on the computing device during a manufacturing operation.

At block 300 the computing device reads an identifier from a keyboard. This identifier can be a serial number or other attribute that uniquely identifies the computing device. Alternatively, in block 300, an identifier is read from an embedded read-only memory element. At block 310, the identifier is transmitted to a

5   network location. This network location can correspond to a Web site on the World Wide Web available by way of the Internet. Alternatively, the network location may be a particular address within a local area network. At block 320, the computing device receives the software package previously installed in the manufacturing environment.

10   FIG. 4 is a block diagram of a method used in a network server that performs network based software recovery in accordance with a preferred embodiment of the invention. Network server 100 of FIG. 1 is suitable for performing the method of FIG. 4. In block 400, a network server receives an identifier from a computing device. At block 410, the network server determines the software package associated

15   with the identifier for a particular computing device. At block 420, the network server assembles the software package previously installed on the computing device during manufacturing and transmits the package by way of a network interface, as in block 430.

In conclusion, network-based software recovery provides a low-cost and

20   efficient means of conveying recovery software to a particular computing device. The method is especially useful for computing devices where a customized software package must be provided to an individual computing device and an image of the software package is loaded on an accompanying readable media, such as a compact disc, in order to permit the user to rebuild the custom software package after the

25   integrity of the data on an internal disk drive has been compromised. Through the use of network-based software recovery, the software package is transmitted to the computing device from a network location, such as a site on the World Wide Web, thus obviating the need to load the custom software package on a readable media for delivery to the user along with the computing device. The method also facilitates the

30   sale of upgraded software products to users.

The foregoing description of the specific embodiments will so fully reveal the general nature of the invention that others can, by applying current knowledge, readily

1. In a computing device, a method for receiving recovery software from a network, comprising:

transmitting an identifier to a network location, said identifier being an attribute of a particular computing device; and

5   receiving recovery software from said network location, said recovery software including a substantial portion of a software package installed during manufacturing of said computing device.

2. The method of claim 1 further comprising receiving said identifier from an

10   input device.

3. The method of claim 2 wherein said input device is a keyboard.

4. The method of claim 1 wherein said identifier is read from a read-only

15   memory.

5. The method of claim 1 wherein said identifier is a serial number of said computing device.

20   6. The method claim 1 wherein said network location is a web site available by way of an interface to the World Wide Web.

7. The method of claim 1 wherein said substantial portion of said software package includes customized software which is not part of a standard software

25   package installed on a model of computing device.

8. The method of claim 1 further comprising receiving software which includes an updated version of said software package installed during manufacturing of said computing device.

30